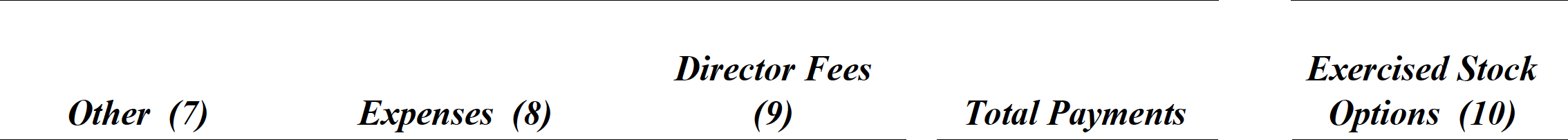
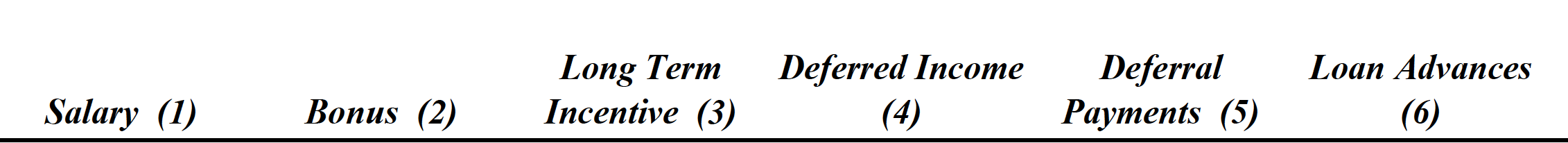
Report on Machine Learning Study

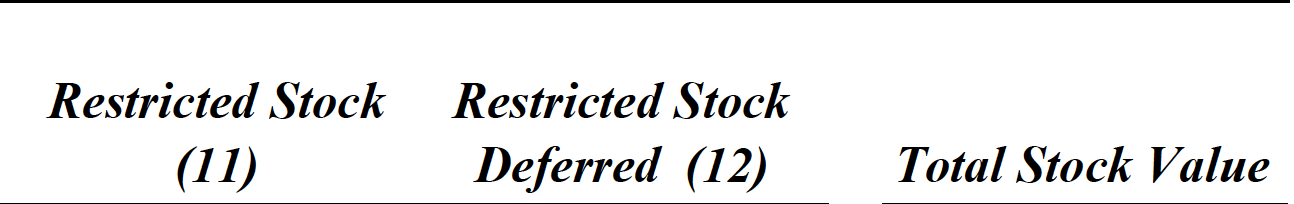
Computer Science has developed into next level with people’s deeper perspective on machine learning, which enables the computer the capacity of ‘thinking’. When people first invented computers, they put them into massive calculation works for it is efficient and time-saving. Because computer never gets tired and always happy to conduct the same and redundant operation.

However, things have changed since people created machine learning algorithm that make it even more precise and efficient for computer to help people make decisions. Therefore, as a university student whoever is interested or majors in Computer Science, it should be necessary for him/her to be familiar with this frontier subfield of Computer Science. Luckily for me, I was instructed by mentor Aaron Tian, and stepped into this new world. Turns out, this course based on machine learning, is quite helpful when understanding the way that computer proceeds and how to deploy those algorithms to do things efficiently.

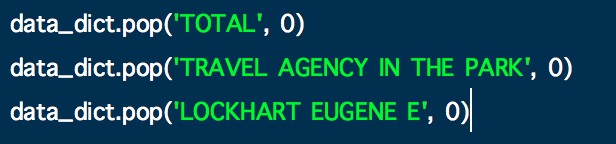
Basically, machine learning is a form of artificial intelligence that allows computer systems to learn from examples, data, and experience. Through enabling computers to perform specific tasks intelligently, machine learning systems can carry out complex processes by learning from data, rather than following pre-programmed rules. The entire course covers Gaussian Naïve Bayes algorithm, support vector machine(SVM), linear regression and so on.

Based on the knowledge learnt, a final project was supposed to be conducted. The goal of this project was to deploy several machine learning techniques, for instance, python sklearn packages, to identify persons of interest (POIs) in the Enron scandal on the basis of the public datasets of emails and finance. The features were identified through the file ‘enron61702insiderpay.pdf’.





From the picture, financial features (such as salary, bonus, and stock options) are listed, therefore, the potential features can be selected. However, outliers in the dataset can result from some malfunction and some data entry errors. Basically utilize the visual techniques without the codes, some outliers can be cleaned as they contain little useful and significant factors. Additionally using the code ‘dictionary.pop(key, 0)’, two key elements are removed



for these are only a spreadsheet tally line for the other 145 data records, some business cannot be considered as persons and the record of him is completely insignificant.

The next task is to create new features, two new features are created from the original features in the dataset and several text features created from the email archives. Start off by testing combinations of features for various algorithms and record the performance of each feature combination to identify the best performing algorithm. Then test again once the best functional algorithm was found. Feature selection and scaling was also performed while creating the testing scenarios for different algorithms. Then store to my dataset for easy export use the “my\_dataset = data\_dict”. To extract features and labels from dataset for local testing, I use the function”

data = featureFormat(my\_dataset, features\_list, sort\_keys = True)

labels, features = targetFeatureSplit(data)”.

Then it is required to try a variety of classifiers. I used Gaussian Naïve Bayes, Support Vector Machine(SVM), Decision Tree Classifier, Random Forest Classifier and Adaboost Classifier.

**No1. Gaussian Naïve Bayes**: Naive Bayes learners and classifiers can be extremely fast compared to more sophisticated methods. The decoupling of the class conditional feature distributions means that each distribution can be independently estimated as a one dimensional distribution. This in turn helps to alleviate problems stemming from the curse of dimensionality.

**No.2 Support Vector Machine(SVM):** Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outliers detection.

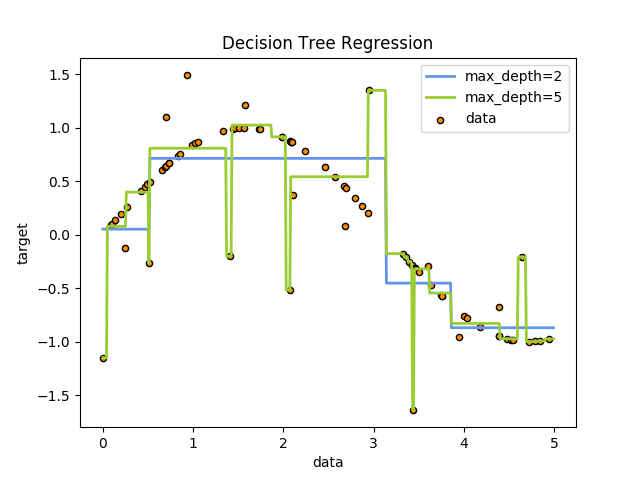
The advantages of support vector machines are:

* Effective in high dimensional spaces.
* Still effective in cases where number of dimensions is greater than the number of samples.
* Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
* Versatile: different Kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

The disadvantages of support vector machines include:

* If the number of features is much greater than the number of samples, avoid over-fitting in choosing Kernel functions and regularization term is crucial.
* SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.

**No.3 Decision Tree Classifier:** Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.



DTs, which does not output that maximum distance from plotted points but certain horizontal or vertical lines, is not like the SVM.

**No.4 Random Forest Classifier :** A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and use averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement if bootstrap=True (default). Actually, as the name of it suggests, it is based on the decision trees with random subsets of observation.

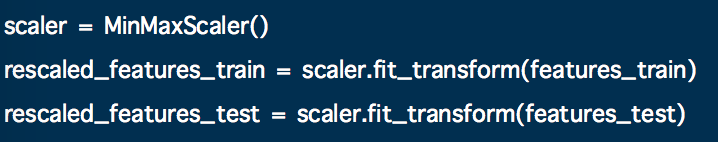
**No.5 KNN:** The principle behind nearest neighbor methods is to find a predefined number of training samples closest in distance to the new point, and predict the label from these. The number of samples can be a user-defined constant (k-nearest neighbor learning), or vary based on the local density of points (radius-based neighbor learning). The distance can, in general, be any metric measure: standard Euclidean distance is the most common choice. Neighbors-based methods are known as non-generalizing machine learning methods, since they simply “remember” all of its training data (possibly transformed into a fast indexing structure such as a [Ball Tree](http://scikit-learn.org/stable/modules/neighbors.html#ball-tree) or [KD Tree](http://scikit-learn.org/stable/modules/neighbors.html#kd-tree).).

Despite its simplicity, nearest neighbors has been successful in a large number of classification and regression problems, including handwritten digits or satellite image scenes. Being a non-parametric method, it is often successful in classification situations where the decision boundary is very irregular.

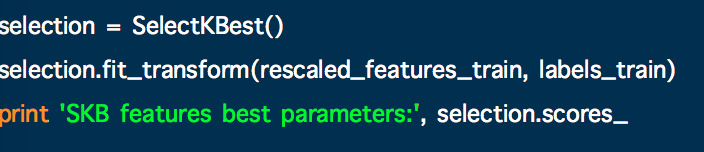
As for tuning part, it is for the computer to learn in the best wat it can, therefore, some parameters can be optimized. And tuning is important for improve the algorithm performance, for instance, the accuracy and recall. In the following is ways of tuning the algorithms. GirdSearchCV is for parameter tuning.



MinMaxScaler is for feature scaling.



SelectKBest is for future selection.



Finally, I use dump\_classifier\_and\_data(clf, my\_dataset, features\_list) to dump the classifiers and datasets and feature\_list so that it can be best tested by tester.py.

To sum up, this final project contains a lot of knowledge of the machine learning. Through using different classifiers and python libraries, the algorithms are efficient and works properly. Although challenges appears occasionally, it is interesting to solve the problems using google. And turns out, it helps me learn a lot.

**References:**

https://saadkhan321.github.io/docs/ProjectReport4\_DataAnalyst\_NanoDegree.pdf

https://en.wikipedia.org/wiki/Machine\_learning

https://royalsociety.org/topics-policy/projects/machine-learning/

http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.\*.html